#4

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN THE APPLICATION OF	:	Ulrich BUNGERT et al.
FOR	:	AUTOMATIC GENERATION OF DIAGNOSTIC PROGRAMS FOR SPS-CONTROLLED SYSTEMS
SERIAL NO.	:	09/972,365
FILED	:	October 5, 2001
EXAMINER	:	Unknown
ART UNIT	:	2184
ATTORNEY DOCKET NO.	:	74953/15381
SIEMENS NO.	:	2001P16810US01

March 12, 2002
Cleveland, Ohio 44115-1475

PRELIMINARY AMENDMENT

Assistant Commissioner of Patents
BOX PATENT APPLICATION
Washington, DC 20231

Dear Sir:

Prior to examination of the present application, please amend the above-identified application as follows:

IN THE DRAWINGS

Please amend Figure 2 accordingly.

IN THE SPECIFICATION

Please insert as the first sentence following the BACKGROUND OF THE INVENTION section of the specification, the following:

--The present invention claims priority from U.S. Provisional Application Ser. No. 60/318,180 entitled "Automatic Generation of Diagnostic Programs for SPS-Controlled Systems" filed September 7, 2001.—

Page 4, line 28, please replace "input" with --inputs--;

Page 8, line 24, please replace "form" with --from--;

Page 8, line 8, please replace "system" with --system, as measured by a supply sensor 221--;

Page 8, line 12, please replace "215" with --214--;

Page 8, line 24, please replace "back" with --rear portion 213--;

Page 8, line 27, please replace "front of" with --front portion 211 of--;

Page 9, line 31, please replace "Jam_back_MV" with --jam_back_MV--;

Page 9, line 31, please replace "valve" with --valve unit 212--;

Page 9, line 32, please replace "valve" with --valve unit 212--;

Page 10, line 1, please replace "front" with --forward--;

Page 10, line 22, please replace "front" with --forward--;

Page 13, line 5, please replace "fp_sensor" with --fp_sensor 278--;

Page 13, line 21, please replace "as" with --and--;

Page 13, line 22 please replace "218" with --220--;

Page 13, line 23 please replace "220" with --218--; and

Page 15, line 31 please replace "settsignal=1" with --setsignal=1--;

IN THE CLAIMS

Please amend claim 1, and add the following new claims 2-19.

1. (Amended) A method of identifying fault conditions in an automation system, comprising the steps of:

identifying components in the system;

identifying inputs and outputs to each identified component, the inputs including possible fault conditions;

determining functional relationships between the inputs and outputs for each identified component; and

determining fault conditions based on the functional relationships and the outputs and other inputs.

2. (New) The method of claim 1, further comprising the step of using the identified inputs and outputs of a specific component and the functional relationships of a corresponding generic component to identify the fault conditions.

3. (New) The method of claim 2, further comprising the step of defining component libraries that describe the functional relationships of the generic components.

4. (New) The method of claim 2, further comprising the step of creating a diagnostic program from the functional relationships of the generic components associated with each component.

5. (New) The method of claim 4, further comprising the step of transforming the functional relationships into fault conditions.

6. (New) The method of claim 5, wherein the step of transforming is implemented in an off-line phase during which the diagnostic program is created, and an on-line phase during which available inputs and outputs are supplied to the transformed functional relationships in the control program, to identify fault conditions.

7. (New) The method of claim 1, further comprising the step of associating a weighting factor with each fault condition.

8. (New) The method of claim 1, further comprising the step of including state information for at least one of the components to define the state of the component at a different time.

9. (New) A method of defining diagnostic code for an automation system, comprising the steps of:

identifying the functional elements of the system;

defining inputs and outputs for each of the functional elements including defining fault conditions as inputs;

defining functional relationships between outputs and associated inputs for each functional element; and

expressing the functional relationships using a programming language.

10. (New) The method of claim 9, wherein the programming language is a symbolic language.

11. (New) The method of claim 9, wherein the step of defining functional relationships for at least some of the functional elements includes utilizing a component library that defines the functional relationships between inputs and outputs of at least one generic element.

12. (New) The method of claim 11, wherein the step of defining the functional relationships includes the step of defining functional relationships and inputs and outputs of the generic elements corresponding to the functional elements in the system.

13. (New) The method of claim 9, further comprising the step of including state information for at least one of the components to define the state of the component at a different time.

14. (New) A method of generating a diagnostic program in an automation system, comprising the steps of:

in an off-line mode, developing a fault model by,

defining a component in the automation system, the component having an input and an output; and

defining a fault relationship between the input and the output of the component, the fault relationship representative of a fault condition;

transforming the fault model into the diagnostic program for execution by an execution device; and

in an on-line mode, executing the diagnostic program in the execution device to identify a component failure of the automation system based upon the fault condition.

15. (New) The method of claim 14, wherein the step of defining a component further comprises the step of defining a plurality of sub-components of the component, each sub-component including an input and output having associated therewith a sub-component fault relationship.

16. (New) The method of claim 14, wherein the fault relationship in the step of defining is developed based upon a component library, which component library describes the functional relationship of generic components.

17. (New) The method of claim 14, further comprising the step of assigning a weighting factor to the fault condition to determine the likelihood that the fault condition is the cause of a component failure.

18. (New) The method of claim 14, wherein the fault condition includes a time dependency such that the corresponding fault relationship describes how a present input value, a present state value, and a present fault cause a present output value and a next state value.

19. (New) The method of claim 14, wherein the execution device is a programmable logic controller also executing a control program that controls the automation system.

REMARKS

Entry of these amendments is respectfully requested. No new matter has been added.

Please charge any additional fees or deficiencies in fees or credit any overpayment to Deposit Account No. 50-0902 of ARTER & HADDEN, LLP, referencing the Docket No. (74953/15381).

Respectfully submitted,

ARTER & HADDEN LLP

Eric D Jorgenson

Eric D. Jorgenson
Registration No. 46,002
1100 Huntington Building
925 Euclid Avenue
(216) 696-2497 (phone)
(216) 696-2645 (fax)

Date: 3/12/2002

626088.1

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to Assistant Commissioner of Patents, Washington, DC 20231.

on MAR. 13, 2002

Valerie A. Salvino
(SIGNATURE)
VALERIE A. SALVINO

1.(Amended) A method of identifying fault conditions in an automation system, comprising the steps of:

identifying components in the system[,];

identifying inputs and outputs to each identified component, the inputs including possible fault conditions[and other inputs,];

determining functional relationships between the inputs and outputs for each identified component[,]; and

determining fault conditions based on the functional relationships and, the outputs and other inputs.